where $T = t_{L-1} - t_0$ is the duration of the bitstream (i.e., the difference between the decoding times for the first and last pictures in the bitstream). And for $R \geq R_N$,

$$B_{min}(R) = B_N \tag{C-31}$$

$$F_{min}(R) = F_N. \tag{C-32}$$

Thus, the leaky bucket parameters can be linearly interpolated and extrapolated.

Alternatively, when the bitstream is communicated to a decoder with buffer size B, it is decodable provided $\geq R_{min}(B)$ and $F \geq F_{min}(B)$, where for $B_n \geq B \geq B_{n+1}$,

$$R_{min}(B) = \alpha R_n + (1 - \alpha)R_{n+1} \tag{C-33}$$

$$F_{min}(B) = \alpha F_n + (1 - \alpha)F_{n+1} \tag{C-34}$$

$$\alpha = (B - B_{n+1}) \div (B_n - B_{n+1}). \tag{C-35}$$

For $B \geq B_1$,

$$R_{min}(B) = R_1 - (B - B_1) \div T \tag{C-36}$$

$$F_{min}(B) = F_1. \tag{C-37}$$

For $B \leq B_N$, the stream may not be decodable.

In summary, the bitstream is guaranteed to be decodable in the sense that the HRD buffer does not overflow or underflow, provided that the point (R,B) lies on or above the lower convex hull of the set of points $(0,B_1+R_1T)$, (R_1,B_1), …, (R_N,B_N), as illustrated in Figure C-6. The minimum start-up delay necessary to maintain this guarantee is $F_{min}(R) \div R$.

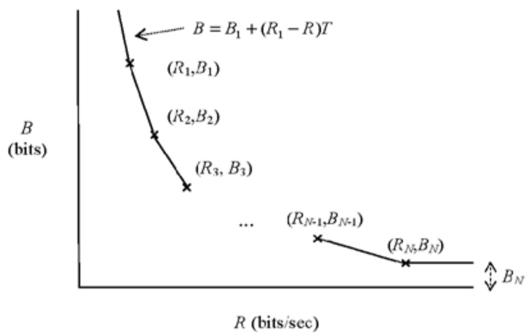

Figure C-6 – Further illustration of the leaky bucket concept

An HRD with buffer size B and initial decoder buffer fullness F with peak input rate R shall perform the tests $B \geq B_{min}(R)$ and $F \geq F_{min}(R)$, as defined above, for any conforming bitstream with LB parameters $(R_1,B_1,F_1),\ldots,(R_N,B_N,F_N)$, and shall decode the bitstream provided that $B \geq B_{min}(R)$ and $F \geq F_{min}(R)$.

C.2.2.4 Encoder considerations

The encoder can create a bitstream that is contained by some given N leaky buckets, or it can simply compute N sets of leaky bucket parameters after the bitstream is generated, or a combination of these. In the former, the encoder enforces the N leaky bucket constraints during rate control. Conventional rate control algorithms enforce only a single leaky bucket constraint. A rate control algorithm that simultaneously enforces N leaky bucket constraints can be obtained by running a conventional rate control algorithm for each of the N leaky bucket constraints, and using as the current quantisation parameter (QP) the maximum of the QP's recommended by the N rate control algorithms.

Additional sets of leaky bucket parameters can always be computed after the fact (whether rate controlled or not), from the bitstream schedule for any given R_n, from the iteration specified in subclause C.2.2.2.

Annex D
Supplemental enhancement information

(This annex forms an integral part of this Recommendation | International Standard)

D.1 Introduction

This annex defines supplemental enhancement information that provides data constructs that are synchronous with the video data content. Each sei_payload( ) defines PayloadType and PayloadSize parameters. This annex defines supplemental enhancement information (SEI) that provides a data delivery mechanism construct that is delivered synchronous with the video data content. SEI assists in the processes related to decoding or display of video. SEI is not required for reconstructing the luma or chroma samples by a video decoder, and decoders are not required to process this information for conformance to this Recommendation | International Standard.

D.2 SEI payload syntax

sei_payload(PayloadType, PayloadSize) {	Category	Descriptor
if(PayloadType = = 1)		
temporal_reference(PayloadSize)	7	
else if(PayloadType = = 2)		
clock_timestamp(PayloadSize)	7	
else if(PayloadType = = 3)		
panscan_rect(PayloadSize)	7	
else if(PayloadType = = 4)		
buffering_period(PayloadSize)	7	
else if(PayloadType = = 5)		
hrd_picture(PayloadSize)	7	
else if(PayloadType = = 6)		
filler_payload(PayloadSize)	7	
else if(PayloadType = = 7)		
user_data_registered_itu_t_t35(PayloadSize)	7	
else if(PayloadType = = 8)		
user_data_unregistered(PayloadSize)	7	
else if(PayloadType = = 9)		
random_access_point(PayloadSize)	7	
else if(PayloadType = = 10)		
ref_pic_buffer_management_repetition(PayloadSize)	7	
else if(PayloadType = = 11)		
spare_picture(PayloadSize)	7	
else if(PayloadType = = 12)		
scene_information(PayloadSize)	7	
else if(PayloadType = = 13)		
subseq_information(PayloadSize)	7	
else if(PayloadType = = 14)		
subseq_layer_characteristics(PayloadSize)	7	
else if(PayloadType = = 15)		
subseq_characteristics(PayloadSize)	7	
else		
reserved_sei_message(PayloadSize)	7	
if(!byte_aligned( )) {		
bit_equal_to_one	7	f(1)
while(!byte_aligned( ))		
bit_equal_to_zero	7	f(1)
}		
}		

D.2.1 Temporal reference SEI message syntax

temporal_reference(PayloadSize) {	Category	Descriptor
progressive_scan	7	u(1)
bottom_field_flag /* zero if progressive_scan is 1 */	7	u(1)
six_reserved_one_bits	7	f(6)
temporal_ref_value	7	u(v)
}		

D.2.2 Clock timestamp SEI message syntax

clock_timestamp(PayloadSize) {	**Category**	**Descriptor**
progressive_scan	7	u(1)
bottom_field_flag /* zero if progressive_scan is 1 */	7	u(1)
six_reserved_one_bits	7	f(6)
counting_type	7	u(5)
full_timestamp_flag	7	u(1)
discontinuity_flag	7	u(1)
count_dropped	7	u(1)
nframes	7	u(8)
if(full_timestamp_flag) {		
seconds_value /* 0,…,59 */	7	u(6)
minutes_value /* 0,…,59 */	7	u(6)
hours_value /* 0,…,23 */	7	u(5)
bit_count = 41		
} else {		
seconds_flag	7	u(1)
bit_count = 25		
if(seconds_flag) {		
seconds_value /* range 0,…,59 */	7	u(6)
minutes_flag	7	u(1)
bit_count += 7		
if(minutes_flag) {		
minutes_value /* 0,…,59 */	7	u(6)
hours_flag	7	u(1)
bit_count += 7		
if(hours_flag) {		
hours_value /* 0,…,23 */	7	u(5)
bit_count += 5		
}		
}		
}		
while(!byte_aligned( )) {		
bit_equal_to_one	7	f(1)
bit_count++		
}		
if(PayloadSize–(bit_count>>3) > 0)		
time_offset	7	i(v)
}		

D.2.3 Pan-scan rectangle SEI message syntax

pan_scan_rect(PayloadSize) {	**Category**	**Descriptor**
pan_scan_rect_id	7	e(v)
pan_scan_rect_left_offset	7	e(v)
pan_scan_rect_right_offset	7	e(v)
pan_scan_rect_top_offset	7	e(v)
pan_scan_rect_bottom_offset	7	e(v)
}		

D.2.4 Buffering period SEI message syntax

buffering_period(PayloadSize) {	**Category**	**Mnemonic**
seq_parameter_set_id	7	ue(v)
if(nal_hrd_flag == 1) {		
for(k = 0; k <= pdb_count; k++)		
initial_pre_dec_removal_delay[k]	7	u(16)
}		
if(vcl_hrd_flag == 1) {		
for(k = 0; k <= pdb_count; k++)		
initial_pre_dec_removal_delay[k]	7	u(16)
}		
prev_buf_period_duration	7	ue(v)
}		

D.2.5 HRD picture SEI message syntax

hrd_picture(PayloadSize)	**Category**	**Descriptor**
pre_dec_removal_delay	7	ue(v)

D.2.6 Filler payload SEI message syntax

filler_payload(PayloadSize) {	**Category**	**Descriptor**
for(k = 0; k < PayloadSize; k++)		
filler_byte	7	f(8) = 0xFF
}		

D.2.7 User data registered by ITU-T Recommendation T.35 SEI message syntax

user_data_registered_itu_t_t35(PayloadSize) {	Category	Descriptor
itu_t_t35_country_code	7	b(8)
if(country_code != 0xFF)		
i = 1;		
else {		
itu_t_t35_country_code_extension_byte	7	b(8)
i = 2;		
}		
do {		
itu_t_t35_payload_byte	7	b(8)
i++		
} while(i < PayloadSize)		
}		

D.2.8 User data unregistered SEI message syntax

user_data_arbitrary(PayloadSize) {	Category	Descriptor
i = 0		
do {		
user_data_arbitrary_payload_byte	7	b(8)
i++		
} while(i < PayloadSize)		
}		

D.2.9 Random access point SEI message syntax

random_access_point(PayloadSize) {	Category	Descriptor
preroll_count	7	ue(v)
postroll_count	7	ue(v)
exact_match_flag	7	u(1)
broken_link_flag	7	u(1)
}		

D.2.10 Reference picture buffer management Repetition SEI message syntax

ref_pic_buffer_management_repetition(PayloadSize) {	Category	Descriptor
original_frame_num	7	u(v)
ref_pic_buffer_management( )		
}		

D.2.11 Spare picture SEI message syntax

spare_picture(PayloadSize) {	Category	Descriptor
delta_frame_num	7	ue(v)
num_spare_pics_minus1	7	ue(v)
for(i = 0; i < num_spare_pics_minus1+1; i++) {		
delta_spare_frame_num	7	ue(v)
ref_area_indicator	7	ue(v)
if(ref_area_indicator = = 1)		
for(j = 0; j < number_of_mbs_in_pic; j++)		
ref_mb_indicator	7	u(1)
else if(ref_area_indicator = = 2) {		
MbCnt = 0		
do {		
zero_run_length	7	ue(v)
MbCnt = MbCnt + zero_run_length + 1		
} while(MbCnt <= MAX_MB_ADDRESS)		
}		
}		

D.2.12 Scene information SEI message syntax

scene_information(PayloadSize) {	Category	Descriptor
scene_id	7	u(8)
scene_transition_type	7	ue(v)
if(scene_transition_type > 3)		
second_scene_id	7	u(8)
}		

D.2.13 Sub-sequence information SEI message syntax

subseq_information(PayloadSize) {	Category	Descriptor
subseq_layer_num	7	ue(v)
subseq_id	7	ue(v)
last_picture_flag	7	u(1)
if(more_sei_payload_data( ))		
stored_frame_cnt	7	ue(v)
}		

D.2.14 Sub-sequence layer characteristics SEI message syntax

subseq_layer_characteristics(PayloadSize) {	Category	Descriptor
do {		
average_bit_rate	7	u(16)
average_frame_rate	7	u(16)
} while(more_sei_payload_data( ))		
}		

D.2.15 Sub-sequence characteristics SEI message syntax

subseq_characteristics(PayloadSize) {	Category	Descriptor
subseq_layer_num	7	ue(v)
subseq_id	7	ue(v)
duration_flag	7	u(1)
if (duration_flag)		
subseq_duration	7	u(32)
average_rate_flag	7	u(1)
if (average_rate_flag) {		
average_bit_rate	7	u(16)
average_frame_rate	7	u(16)
}		
num_referenced_subseqs	7	ue(v)
for (n = 0; n < num_referenced_subseqs; n++) {		
ref_subseq_layer_num	7	ue(v)
ref_subseq_id	7	ue(v)
}		
}		

D.2.16 Reserved SEI message syntax

reserved_sei_message(PayloadSize) {	Category	Descriptor
for(i=0; i<PayloadSize; i++)		
reserved_sei_message_payload_byte	7	b(8)
}		

D.3 SEI payload semantics

D.3.1 Temporal reference SEI message semantics

progressive_scan: This parameter indicates whether the current picture has progressive or interlaced scan timing.

bottom_field_flag: When progressive_scan is 0, this parameter indicates whether the temporal reference is for the top (0) or bottom (1) field. Shall be 0 if progressive_scan is 1.

six_reserved_one_bits: Reserved for future backward-compatible use by ITU-T | ISO/IEC. Shall be equal to the binary string '111111' unless and until specified otherwise by ITU-T | ISO/IEC. A decoder conforming to this Recommendation | International Standard shall ignore the value of these bits.

temporal_ref_value: This parameter indicates a number of clock ticks as a multiplier of num_units_in_tick for the current time scale. It is used for conveying local relative timing information.

The number of bytes used by temporal_ref_value shall remain constant for the video stream and shall be equal to PayloadSize – 1 bytes. For a temporal_ref_value encoded using *n* bytes, the temporal_reference contains the remainder of a clock tick counter modulo 256^n.

D.3.2 Clock timestamp SEI message semantics

The contents of the clock timestamp SEI message specify a time_offset which indicates the display or capture time computed as

$$\text{equivalent\_timestamp} = ((HH * 60 + MM) * 60 + \text{SS}) * \text{time\_scale} + NF * \text{num\_units\_in\_tick} + TO, \quad \text{(D-1)}$$

in units of ticks of a clock with clock frequency equal to time_scale Hz.

progressive_scan: This parameter indicates whether the current picture is in progressive or interlaced scan format.

bottom_field_flag: When progressive_scan is 0, this parameter indicates whether the temporal reference is for the top (0) or bottom (1) field. Shall be 0 if progressive_scan is 1.

six_reserved_one_bits: Reserved for future use by ITU-T | ISO/IEC. Shall be equal to the binary string '111111'. A decoder conforming to this Recommendation | International Standard shall ignore the value of these bits.

counting_type: A 5-bit parameter that specifies the method of dropping values of the nframes parameter as defined in Table D-1.

Table D-1 – Definition of counting_type values

Value (binary)	Interpretation
00000	no dropping of nframes count values and no use of time_offset
00001	no dropping of nframes count values
00010	dropping of individual zero values of nframes count
00011	dropping of individual max_pps values of npictures count
00100	dropping of the two lowest (value 0 and 1) nframes counts when seconds_value is zero and minutes_value is not an integer multiple of ten
00101	dropping of unspecified individual nframes count values
00110	dropping of unspecified numbers of unspecified nframes count values
00111 - 11111	reserved

full_timestamp_flag indicates whether the nframes parameter is followed by seconds_value or seconds_flag.

discontinuity_flag indicates whether the time difference between the current value of equivalent_timestamp and the value of equivalent_timestamp computed from the last previously-transmitted clock timestamp can be interpreted as a true time difference. A value of 0 indicates that the difference represents a true time difference.

count_dropped indicates the skipping of a count using the counting method specified by counting_type.

nframes indicates the value of *NF* used to compute the equivalent_timestamp. Shall be less than

$$\text{max\_fps} = \text{Ceil}\,(\,\text{time\_scale} \div \text{num\_units\_in\_tick}\,) \quad \text{(D-2)}$$

If counting_type is '00010' and count_dropped is 1, nframes shall be 1 and the value of nframes for the last previous picture in display order shall not be equal to 0 unless discontinuity_flag is equal to 1.

If counting_type is '00011' and count_dropped is 1, nframes shall be 0 and the value of nframes for the last previous picture in display order shall not be equal to max_fps – 1 unless discontinuity_flag is equal to 1.

If counting_type is '00100' and count_dropped is 1, nframes shall be 2 and the indicated value of *SS* shall be zero and the indicated value of *MM* shall not be an integer multiple of ten and nframes for the last previous picture in display order shall not be equal to 0 or 1 unless discontinuity_flag is equal to 1.

If counting_type is '00101' or '110' and count_dropped is 1, nframes shall not be equal to one plus the value of nframes for the last previous picture in display order modulo max_fps unless discontinuity_flag is equal to 1.

seconds_flag indicates whether seconds_value is present when full_timestamp_flag is 0.

seconds_value indicates the value of *SS* used to compute the equivalent_timestamp. Shall not exceed 59. If not present, the last previously-transmitted seconds_value shall be used as *SS* to compute the equivalent_timestamp.

minutes_flag indicates whether seconds_value is present when full_timestamp_flag is 0 and seconds_flag is 1.

minutes_value indicates the value of *MM* used to compute the equivalent_timestamp. Shall not exceed 59. If not present, the last previously-transmitted minutes_value shall be used as *MM* to compute the equivalent_timestamp.

hours_flag indicates whether seconds_value is present when full_timestamp_flag is 0 and seconds_flag is 1 and minutes_flag is 1.

hours_value indicates the value of *HH* used to compute the equivalent_timestamp. Shall not exceed 23. If not present, the last previously-transmitted hours_value shall be used as *HH* to compute the equivalent_timestamp.

bit_equal_to_one is a single bit which shall be equal to 1.

time_offset indicates the value of *TO* used to compute the equivalent_timestamp. The number of bytes used to represent time_offset shall be equal to PayloadSize – (bit_count >> 3), where bit_count is computed as specified in subclause D.2.2. If time_offset is not present, the value 0 shall be used as *TO* to compute the equivalent_timestamp.

D.3.3 Pan-scan rectangle SEI message semantics

The pan-scan rectangle SEI message parameters define the coordinates of a rectangle relative to the cropping rectangle of the picture parameter set. Each coordinate of this rectangle is defined in units of $1/16^{th}$ sample spacing relative to the luma sampling grid.

pan_scan_rect_id contains an identifying number which may be used as specified externally to identify the purpose of the pan-scan rectangle (for example, to identify the rectangle as the area to be shown on a particular display device or as the area that contains a particular actor in the scene).

pan_scan_rect_left_offset, pan_scan_rect_right_offset, pan_scan_rect_top_offset, and **pan_scan_rect_bottom_offset** specify, as signed integer quantities in units of $1/16^{th}$ sample spacing relative to the luma sampling grid, the location of the pan-scan rectangle.

The pan-scan rectangle is defined, in units of $1/16^{th}$ sample spacing relative to the luma sampling grid, as the area of the rectangle with horizontal coordinates from 16 * cropping_rect_left + pan_scan_rect_left_offset to 16 * [16 * (pic_width_in_mbs_minus1 + 1) – cropping_rect_right] + pan_scan_rect_right_offset – 1 and with vertical coordinates from 16 * cropping_rect_top + pan_scan_rect_top_offset to 16 * [16 * (pic_height_in_mbs_minus1 + 1) – cropping_rect_bottom] + pan_scan_rect_bottom_offset – 1, inclusive. If this rectangular area includes samples outside of the cropping rectangle, the region outside of the cropping rectangle may be filled with synthesized content (such as black video content or neutral grey video content) for display.

D.3.4 Buffering period SEI message semantics

A Buffering Period is defined as the set of pictures between two instances of the Buffering Period SEI message. The seq_parameter_set_id indicates the sequence parameter set that contains the sequence level HRD attributes.

seq_parameter_set_id indicates the sequence parameter set that contains the sequence level HRD attributes.

initial_pre_dec_removal_delay: This syntax element represents the delay between the time of arrival in the pre-decoder buffer of the first bit of the coded data associated with the first picture following the Buffering Period SEI message (including all NAL data in the case that the HRD pertains to the NAL) and the time of removal of the coded data associated with the picture from the pre-decoder buffer. It is in units of a 90 kHz clock. The initial_pre_dec_removal_delay syntax element is used in conjunction with the pre-decoder buffers as specified in Annex C. A value of zero is forbidden.

prev_buf_period_duration: This syntax element represents the duration of the subset of the video sequence contained in the previous Buffering Period. The interpretation of the syntax element is as a number of clock ticks (see Annex D). The prev_buf_period_duration syntax element is used in conjunction with the pre-decoder buffers as specified in Annex C. A value of zero is forbidden.

D.3.5 HRD picture SEI message semantics

pre_dec_removal_delay: This syntax element indicates how many clock ticks (see Annex C) to wait after removal from the HRD pre-decoder buffer of the previous picture before removing from the buffer the picture data immediately following the SEI message which contains the element. This value is also used to calculate an earliest possible time of arrival of picture data into the pre-decoder buffer, as defined in Annex C.

D.3.6 Filler payload SEI message semantics

This message contains a series of PayloadSize bytes of value 0xFF, which can be discarded.

filler_byte shall be a byte having the value 0xFF.

D.3.7 User data registered by ITU-T Recommendation T.35 SEI message semantics

This message contains registered user data as specified by ITU-T Recommendation T.35.

itu_t_t35_country_code shall be a byte having a value specified as a country code by ITU-T Recommendation T.35.

itu_t_t35_country_code_extension_byte shall be a byte having a value specified as an extended country code by ITU-T Recommendation T.35.

itu_t_t35_payload_byte shall be a byte containing user data registered as specified by ITU-T Recommendation T.35.

D.3.8 User data arbitrary SEI message semantics

This message contains arbitrary user data, the contents of which are not specified by this Recommendation | International Standard.

user_data_arbitrary_payload_byte shall be a byte having a value not specified by this Recommendation | International Standard.

NOTE - Users of this Recommendation | International Standard should exercise care in the use of the user data arbitrary SEI message to avoid the carriage of data content in a form likely to conflict with the data content format of other users (e.g., avoiding conflict by using a fixed multi-byte prefix identifier within the payload content).

D.3.9 Random access point SEI message semantics

The random access point SEI message indicates the recovery point of decoder output after starting decoding from a random access entry point. All decoded pictures at or subsequent to the recovery point in output order are indicated to be correct or approximately correct in content. Decoded pictures produced by starting the decoding process at the entry point may not be correct in content until the indicated recovery point, and the decoding process starting at the entry point and ending at the recovery point may contain references to pictures not available in the multi-picture buffer.

The entry point is indicated as a pre-roll count relative to the position of the SEI message in units of coded frame numbers prior to the frame number of the current picture. The recovery point is indicated as a post-roll count in units of coded pictures subsequent to the current picture at the position of the SEI message.

preroll_count indicates the entry point for the decoding process. Decoding should have started at or prior to the stored picture having the frame number equal to the frame number of the next slice minus the preroll_count in modulo MAX_FN arithmetic.

postroll_count indicates the recovery point of output. All decoded pictures in output order are indicated to be correct or approximately correct in content after the stored picture having the frame number equal to the frame number of the next slice incremented by postroll_count in modulo MAX_FN arithmetic.

exact_match_flag indicates whether decoded pictures at and subsequent to the recovery point in output order obtained by starting the decoding process at the specified entry point shall be an exact match to the pictures that would be produced by a decoder starting at the last prior IDR point in the NAL unit stream. The value 0 indicates that the match may not be exact and the value 1 indicates that the match shall be exact.

If decoding starts from an entry point indicated in a random access point SEI message, all references to unavailable stored pictures shall be inferred as references to sample values given by Y=Cb=Cr=128 (mid-level grey) for purposes of determining the conformance of the value of exact_match_flag.

broken_link_flag indicates the presence or absence of a splicing point in the NAL unit stream at the location of the random access point SEI message. If broken_link_flag is equal to 1, pictures produced by starting the decoding process at the last previous IDR point may contain undesirable visual artifacts due to splicing operations and should not be

displayed until the indicated random access recovery point in output order. If broken_link_flag is equal to 0, no indication is given regarding any potential presence of visual artifacts.

If a sub-sequence information SEI message is transmitted in conjunction with a random access point SEI message in which broken_link_flag is equal to 1 and if subseq_layer_num is 0, subseq_id should be different from the latest subseq_id for subseq_layer_num equal to 0 that was decoded prior to the entry point. If broken_link_flag is equal to 0, the subseq_id in sub-sequence layer 0 should remain unchanged.

A buffering period SEI message should be transmitted at the location of the random access entry point indicated in the random access point SEI message in order to establish initialisation of the HRD buffer model.

D.3.10 Reference picture buffer management Repetition SEI message semantics

The Reference picture buffer management repetition SEI message is used to repeat memory management control operation commands that were located earlier in decoding order.

original_frame_num identifies the picture were the repeated memory management control operation originally occurred.

ref_pic_buffer_managament( ) shall contain a copy of the reference picture buffer management syntax elements of the picture whose frame_num was original_frame_num.

D.3.11 Spare picture SEI message semantics

The spare picture SEI message indicates that certain macroblocks, called spare decoded macroblocks, in one or more decoded stored pictures resemble the co-located macroblocks in a certain decoded picture, called the target picture, so much that any of these spare decoded macroblocks can be used to replace a co-located incorrect decoded macroblock in the target picture in the multi-frame buffer and in decoder output. Decoded pictures that contain spare macroblocks are called spare pictures.

The picture that contains the next slice or data partition in decoding order is herein referred to as the current picture. The frame_num of the current picture is herein denoted as CurrFrameNum.

delta_frame_num identifies the target picture whose spare pictures and macroblocks are specified later in the message. Let TargetFrameNum be the frame_num of the target picture, and the target picture is the stored picture having the TargetFrameNum. TargetFrameNum is calculated as follows

$$\begin{aligned}&\text{TargetFrameNum} = \text{CurrFrameNum} - \text{delta\_frame\_num}\\&\text{if(TargetFrameNum} < 0\text{)}\\&\qquad\text{TargetFrameNum} = \text{MAX\_FN} + \text{TargetFrameNum}\end{aligned} \tag{D-3}$$

num_spare_pics_minus1 specifies the number of pictures which contain spare picture or macroblocks for the target picture.

delta_spare_frame_num specifies to which spare picture the following spare picture information in the current loop count belongs. For the first spare picture of the message, CandidateSpareFrameNum is equal to TargetFrameNum – 1 if TargetFrameNum is greater than 0 and MAX_FN – 1 otherwise. For later spare pictures, CandidateSpareFrameNum is the SpareFrameNum of the previous loop round minus 1 if SpareFrameNum is greater than 0 and MAX_FN – 1 otherwise. For each loop round, SpareFrameNum is calculated as follows:

$$\begin{aligned}&\text{SpareFrameNum} = \text{CandidateSpareFrameNum} - \text{delta\_spare\_frame\_num}\\&\text{if(SpareFrameNum} < 0\text{)}\\&\qquad\text{SpareFrameNum} = \text{MAX\_FN} + \text{SpareFrameNum}\end{aligned} \tag{D-4}$$

ref_area_indicator specifies how the locations of spare macroblocks are coded. ref_area_indicator 0 indicates that all macroblocks of the spare picture are spare macroblocks.. ref_area_indicator 1 indicates an uncompressed spare macroblock map. ref_area_indicator 2 indicates a compressed spare macroblock map. A spare macroblock map consists of flags for each macroblock location of a picture. A flag shall be 0 if the macroblock location in the spare picture is a spare macroblock and 1 otherwise.

If ref_area_indicator is 1, there is a **ref_mb_indicator** for each macroblock address of the spare macroblock map in raster scan order. ref_mb_indicator 0 indicates that the macroblock is a spare macroblock, and ref_mb_indicator 1 indicates that the macroblock is not a spare macroblock.

If ref_area_indicator is 2, a spare macroblock map between a spare picture and the target picture is compressed. The coded macroblock map for loop_count equal to 0 is the spare macroblock map between the target picture and the first spare picture. A coded macroblock map for loop_count greater than 0 is generated by applying an exclusive or operation between the previous spare macroblock map and the current spare macroblock map. The coded macroblock map is

scanned in counter-clockwise box-out order as specified in subclause 8.3.4.1. The number of consecutive zeros in the scanning order is indicated in **zero_run_length**.

D.3.12 Scene information SEI message semantics

A scene is herein defined as a set of pictures in decoding order captured with one camera. The scene information SEI message is used to label scenes with identifiers. The message concerns the next slice or data partition in decoding order.

scene_id: Pictures in a scene shall share the same value of scene_id. Consecutive scenes in decoding order should not have the same value of scene_id. If the next slice or data partition in decoding order belongs to a picture that includes contents from two scenes, scene_id is the scene identifier of the former scene in decoding order.

The following values of **scene_transition_type** are valid:

Table D-2 – Scene transition types.

Value	Description
0	No transition
1	Fade-out
2	Fade-in
3	Unspecified transition from or to constant color
4	Dissolve
5	Wipe
6	Unspecified mixture of two scenes
Other values	Reserved

If scene transition type is greater than 3, the next slice or data partition in decoding order belongs to a picture that includes contents from two scenes.

second_scene_id is present if the next slice or data partition in decoding order belongs to a picture that includes contents from two scenes. second_scene_id is the scene identifier of the latter scene in decoding order.

D.3.13 Sub-sequence information SEI message semantics

The sub-sequence information SEI message is used to indicate the position of a picture in data dependency hierarchy that consists of sub-sequence layers and sub-sequences.

A sub-sequence layer contains a subset of the coded pictures in a coded data stream. Sub-sequence layers are numbered with non-negative integers. A layer having a larger layer number is a higher layer than a layer having a smaller layer number. The layers are ordered hierarchically based on their dependency on each other so that a layer does not depend on any higher layer and may depend on lower layers. In other words, layer 0 is independently decodable, pictures in layer 1 may be predicted from layer 0, pictures in layer 2 may be predicted from layers 0 and 1, etc. The subjective quality increases along with the number of decoded layers.

A sub-sequence is a set of coded pictures within a sub-sequence layer. A picture shall reside in one sub-sequence layer and in one sub-sequence only. A sub-sequence shall not depend on any other sub-sequence in the same or in a higher sub-sequence layer. A sub-sequence in layer 0 can be decoded independently of any other sub-sequences and previous long-term reference pictures.

The sub-sequence information SEI message concerns the next slice or data partition in decoding order. The picture which the next slice or data partition belongs to is herein referred to as the target picture.

subseq_layer_num indicates the sub-sequence layer number of the target picture.

subseq_id identifies the sub-sequence within a layer. Consecutive sub-sequences within a particular layer in decoding order shall have a different subseq_id from each other.

last_picture_flag equal to 1 signals that the target picture is the last picture of the sub-sequence (in decoding order).

stored_frame_cnt is 0 for the first stored picture of the sub-sequence. For each coded frame belonging to the sub-sequence in decoding order, stored_frame_cnt shall be incremented by 1, in modulo MAX_FN operation, relative to the previous stored frame that belongs to the sub-sequence.

D.3.14 Sub-sequence layer characteristics SEI message semantics

The sub-sequence layer characteristics SEI message indicates the characteristics of sub-sequence layers.

A pair of average bit rate and average frame rate characterizes each sub-sequence layer. The first pair of average bit rate and average frame rate signals the characteristics of sub-sequence layer 0. The second pair, if present, signals the characteristics of sub-sequence layers 0 and 1 jointly. Each pair in decoding order signals the characteristics for a range of sub-sequence layers from layer number 0 to the layer number that is incremented by one from the previous upper limit of layer numbers. The values are in effect from the point they are decoded until an update of the values is decoded.

average_bit_rate gives the average bit rate in units of 1000 bits per second. All NAL units in the range of sub-sequence layers specified above are taken into account in the calculation. The average bit rate is calculated according to the decoding time of the NAL units. Value zero means an undefined bit rate.

average_frame_rate gives the average frame rate in frames/(256 seconds) of the sub-sequence layer. Value zero indicates an undefined frame rate.

D.3.15 Sub-sequence characteristics SEI message semantics

The sub-sequence characteristics SEI message indicates the characteristics of a sub-sequence. It also indicates inter prediction dependencies between sub-sequences.

This message applies to the next sub-sequence in decoding order having the indicated **subseq_layer_num** and **subseq_id**. This sub-sequence is herein called the target sub-sequence.

duration_flag equal to zero indicates that the duration of the target sub-sequence is not specified.

subseq_duration indicates the duration of the target sub-sequence in clock ticks of a 90-kHz clock.

average_rate_flag equal to zero indicates that the average bit rate and the average frame rate of the target sub-sequence are unspecified.

average_bit_rate gives the average bit rate in (1000 bits)/second of the target sub-sequence. All NAL units of the target sub-sequence are taken into account in the calculation. The average bit rate is calculated according to the decoding time of the NAL units.

average_frame_rate gives the average frame rate in frames/(256 seconds) of the current sub-sequence.

num_referenced_subseqs gives the number of sub-sequences which contain pictures that are used as reference pictures for inter prediction in the pictures of the target sub-sequence.

ref_subseq_layer_num and **ref_subseq_id** identify a sub-sequence that contains pictures that are used as reference pictures for inter prediction in the pictures of the target sub-sequence.

D.3.16 Reserved SEI message semantics

This message consists of data reserved for future backward-compatible use by ITU-T | ISO/IEC. Encoders conforming to this Recommendation | International Standard shall not send reserved SEI messages until and unless the use of such messages has been specified by ITU-T | ISO/IEC. Decoders conforming to this Recommendation | International Standard that encounter reserved SEI messages shall discard their content without effect on the decoding process, except as specified in future Recommendations | International Standards defined by ITU-T | ISO/IEC.

reserved_sei_message_payload_byte is a byte reserved for future use by ITU-T | ISO/IEC.

Annex E
Video usability information

(This annex forms an integral part of this Recommendation | International Standard)

E.1 Introduction

This Annex specifies those parts of the sequence parameter set and the picture parameter set that are not required for determining the decoded values of samples. The parameters specified in this annex can be used to facilitate the use of the decoded pictures or facilitate the resource allocation of a decoder by restricting certain video parameters beyond those limits specified by Annex A. Decoders are not required to process VUI sequence parameters for conformance to this Recommendation | International Standard.

For each of the parameters of this Annex, default values are defined in the semantics subclause. The syntax includes flags that allow avoiding the signalling of groups of parameters. If a specific group of parameters is not coded, the default values for the parameters become effective.

E.2 VUI syntax

E.2.1 VUI sequence parameters syntax

vui_seq_parameters( ) {	Category	Descriptor
aspect_ratio_info_flag	0	u(1)
if(aspect_ratio_info_flag) {		
aspect_ratio_info	0	b(8)
if(aspect_ratio_info = = "Extended SAR") {		
sar_width	0	u(8)
sar_height	0	u(8)
}		
}		
video_signal_type_flag	0	u(1)
if(video_signal_type_flag) {		
video_format	0	u(3)
video_range_flag	0	u(1)
colour_description_flag	0	u(1)
if(colour_description_flag) {		
colour_primaries	0	b(8)
transfer_characteristics	0	b(8)
matrix_coefficients	0	b(8)
}		
}		
chroma_location_flag	0	u(1)
if (chroma_location_flag) {		
chroma_location_frame	0	e(v)
chroma_location_field	0	e(v)
}		
timing_information_flag		
if(timing_information_flag) {		
num_units_in_tick	0	u(32)
time_scale	0	u(32)
fixed_frame_rate_flag	0	u(1)
}		
nal_hrd_flag	0	u(1)
if(**nal_hrd_flag** = = 1)		
hrd_parameters( )		
vcl_hrd_flag	0	u(1)
if(**vcl_hrd_flag** = = 1)		
hrd_parameters( )		
if((**nal_hrd_flag** = = 1 || (**vcl_hrd_flag** = = 1)) {		
low_delay_hrd	0	u(1)
removal_time_tolerance	0	ue(v)
}		
bitstream_restriction_flag	0	u(1)
if(bitstream_restriction_flag) {	0	u(1)
motion_vectors_over_pic_boundaries_flag	0	u(1)
minimum_compression_per_pic_reversed	0	e(v)

minimum_compression_per_macroblock_reversed	0	e(v)
log2_maximum_mv_length_vertical	0	e(v)
log2_maximum_mv_length_horizontal	0	e(v)
}		
}		

E.2.2 HRD parameters syntax

hrd_parameters( ) { /* coded picture buffer parameters */		
pdb_cnt	0	ue(v)
bit_rate_scale	0	u(4)
coded_pic_buffer_size_scale	0	u(4)
for(k=1; k<=pdb_cnt; k++) {		
bit_rate_value[k]	0	ue(v)
coded_pic_buffer_size_value[k]	0	ue(v)
vbr_cbr_flag[k]	0	u(1)
}		
}		

E.2.3 VUI picture parameters syntax

vui_pic_parameters( ) {	**Category**	**Descriptor**
frame_cropping_flag	1	ue(v)
if(frame_cropping_flag) {		
frame_cropping_rect_left_offset	1	ue(v)
frame_cropping_rect_right_offset	1	ue(v)
frame_cropping_rect_top_offset	1	ue(v)
frame_cropping_rect_bottom_offset	1	ue(v)
}		
}		

E.3 VUI semantics

E.3.1 VUI sequence parameters semantics

aspect_ratio_info_flag: A flag that, when 1, signals the presence of the aspect_ratio_info. If the flag is 0, then the following default values shall apply: aspect_ratio_info = 0.

aspect_ratio_info is an eight-bit integer which defines the value of sample aspect ratio. Table E-1 shows the meaning of the code. If aspect_ratio_info indicates Extended SAR, sample_aspect_ratio is represented by sar_width and sar_height. The sar_width and sar_height shall be relatively prime. If aspect_ratio_info is zero or if either sar_width or sar_height are zero, the sample aspect ratio shall be considered unspecified or specified externally.

Table E-1 – Meaning of sample aspect ratio

aspect_ratio_info	Sample aspect ratio
0000 0000	Undefined or specified externally
0000 0001	1:1 ("Square")
0000 0010	12:11 (625-type for 4:3 picture)
0000 0011	10:11 (525-type for 4:3 picture)
0000 0100	16:11 (625-type stretched for 16:9 picture)
0000 0101	40:33 (525-type stretched for 16:9 picture)

0000 0110	24:11 (Half-wide 4:3 for 625)
0000 0111	20:11 (Half-wide 4:3 for 525)
0000 1000	32:11 (Half-wide 16:9 for 625)
0000 1001	80:33 (Half-wide 16:9 for 525)
0000 1010	18:11 (2/3-wide 4:3 for 625)
0000 1011	15:11 (2/3-wide 4:3 for 525)
0000 1100	24:11 (2/3-wide 16:9 for 625)
0000 1101	20:11 (2/3-wide 16:9 for 525)
0000 1110	16:11 (3/4-wide 4:3 for 625)
0000 1111	40:33 (3/4-wide 4:3 for 525)
0001 0000	64:33 (3/4-wide 16:9 for 625)
0001 0001	160:99 (3/4-wide 16:9 for 525)
0001 0010 to 1111 1110	Reserved
1111 1111	Extended SAR

sar_width is an 8-bit unsigned integer which indicates the horizontal size of sample aspect ratio. A zero value is forbidden.

sar_height is an 8-bit unsigned integer which indicates the vertical size of sample aspect ratio. A zero value is forbidden.

video_signal_type_flag: A flag that, when 1, signals the presence of video signal information. If video_signal_type_flag is 0, then the following default values shall apply: video_format = '101', video_range = 0, colour_description = 0.

video_format: This is a three bit integer indicating the representation of the pictures before being coded in accordance with this Recommendation | International Standard. Its meaning is defined in Table E-2. If the video_signal_type( ) is not present in the bitstream then the video format may be assumed to be "Unspecified video format".

Table E-2 – Meaning of video_format

video_format	Meaning
000	Component
001	PAL
010	NTSC
011	SECAM
100	MAC
101	Unspecified video format
110	Reserved
111	Reserved

video_range_flag indicates the nominal black level and range of the luminance and chrominance signals as derived from E'_Y, E'_{PB}, and E'_{PR} analogue component signals as follows:

If video_range_flag=0:

$$Y = \text{round}(219 * E'_Y + 16)$$
$$Cb = \text{round}(224 * E'_{PB} + 128)$$
$$Cr = \text{round}(224 * E'_{PR} + 128)$$

If video_range_flag=1:

$$Y = \text{round}(255 * E'_Y)$$
$$Cb = \text{round}(255 * E'_{PB} + 128)$$
$$Cr = \text{round}(255 * E'_{PR} + 128)$$

If video_signal_type_flag is zero, video_range shall be inferred to have value 0 (a nominal range of Y from 16 to 235).

colour_description_flag which if set to '1' indicates the presence of colour_primaries, transfer_characteristics and matrix_coefficients in the bitstream.

colour_primaries: This 8-bit integer describes the chromaticity coordinates of the source primaries, and is defined in Table E-3.

Table E-3 – Colour Primaries

Value	Primaries
0	Reserved
1	ITU-R Recommendation BT.709 primary x y green 0,300 0,600 blue 0,150 0,060 red 0,640 0,330 white D65 0,3127 0,3290
2	Unspecified video Image characteristics are unknown.
3	Reserved
4	ITU-R Recommendation BT.470-2 System M primary x y green 0,21 0,71 blue 0,14 0,08 red 0,67 0,33 white C 0,310 0,316
5	ITU-R Recommendation BT.470-2 System B, G primary x y green 0,29 0,60 blue 0,15 0,06 red 0,64 0,33 white D65 0,3127 0,3290
6	SMPTE 170M primary x y green 0,310 0,595 blue 0,155 0,070 red 0,630 0,340 white D65 0,3127 0,3290
7	SMPTE 240M (1987) primary x y green 0,310 0,595 blue 0,155 0,070 red 0,630 0,340 white D65 0,3127 0,3290
8	Generic film (colour filters using Illuminant C) primary x y green 0,243 0,692 (Wratten 58) blue 0,145 0,049 (Wratten 47) red 0,681 0,319 (Wratten 25)
9-255	Reserved

If video_signal_type_flag is zero or colour_description is zero, the chromaticity is unspecified or specified externally.

transfer_characteristics: This 8-bit integer describes the opto-electronic transfer characteristic of the source picture, and is defined in Table E-4.

Table E-4 – Transfer Characteristics

Value	Transfer Characteristic
0	Reserved

1	ITU-R Recommendation BT.709 $V = 1{,}099\ L_c^{0{,}45} - 0{,}099$ for $1 \geq L_c \geq 0{,}018$ $V = 4{,}500\ L_c$ for $0{,}018 > L_c \geq 0$
2	Unspecified video Image characteristics are unknown.
3	Reserved
4	ITU-R Recommendation BT.470-2 System M Assumed display gamma 2,2
5	ITU-R Recommendation BT.470-2 System B, G Assumed display gamma 2,8
6	SMPTE 170M $V = 1{,}099\ L_c^{0{,}45} - 0{,}099$ for $1 \geq L_c \geq 0{,}018$ $V = 4{,}500\ L_c$ for $0{,}018 > L_c \geq 0$
7	SMPTE 240M (1987) $V = 1{,}1115\ L_c^{0{,}45} - 0{,}1115$ for $L_c \geq 0{,}0228$ $V = 4{,}0\ L_c$ for $0{,}0228 > L_c$
8	Linear transfer characteristics i.e. $V = L_c$
9	Logarithmic transfer characteristic (100:1 range) $V = 1.0 - \log_{10}(Lc)/2$ for $1 = L_c = 0.01$ $V = 0.0$ for $0.01 > L_c$
10	Logarithmic transfer characteristic (316.22777:1 range) $V = 1.0 - \log_{10}(Lc)/2.5$ for $1 = L_c = 0.0031622777$ $V = 0.0$ for $0.0031622777 > L_c$
11-255	Reserved

If video_signal_type_flag is zero or colour_description is zero, the transfer characteristics are unspecified or are specified externally.

matrix_coefficients: This 8-bit integer describes the matrix coefficients used in deriving luminance and chrominance signals from the green, blue, and red primaries, as specified in Table E-5.

Using this table:

E'_Y is analogue with values between 0 and 1
E'_R, E'_G, and E'_B are analogue with values between 0 and 1
E'_{PB} and E'_{PR} are analogue between the values -0,5 and 0,5
White is defined as $E'_R - E'_G - E'_B - 1$
White equivalently given by $E'_Y = 1$, $E'_{PB} = 0$, $E'_{PR} = 0$
$E'_Y = K_R * E'_R + (1 - K_R - K_B) * E'_G + K_B * E'_B$
$E'_{PB} = 0.5(E'_R - E'_Y) \div (1-K_R)$
$E'_{PB} = 0.5 (E'_B \quad E'_Y) \div (1-K_B)$

Table E-5 – Matrix Coefficients

Value	Matrix
0	Reserved
1	ITU-R Recommendation BT.709 $K_G = 0{,}7152$; $K_R = 0{,}2126$
2	Unspecified video Image characteristics are unknown.
3	Reserved
4	FCC $K_G = 0{,}59$; $K_R = 0{,}30$
5	ITU-R Recommendation BT.470-2 System B, G: $K_G = 0{,}587$; $K_R = 0{,}299$
6	SMPTE 170M $K_G = 0{,}587$; $K_R = 0{,}299$
7	SMPTE 240M (1987) $K_G = 0{,}701$; $K_R = 0{,}212$
8-255	Reserved

If video_signal_type_flag is zero or colour_description is zero, the matrix coefficients are assumed to be undefined or specified externally.

chroma_location_flag: A flag that, when 1, signals the presence of the chroma location information. If the flag is 0, then the following default values shall apply: chroma_location_frame = 0, chroma_location_field = 0.

chroma_location_frame specifies the 4:2:0 sampling structure according to Table E-6 and Figure E-1.

Table E-6 – Chroma Sampling Structure Frame

Value	Sampling Structure
0	undefined
1	Frame according to Figure E-1 Chroma Sample Mode 1
2	Frame according to Figure E-1 Chroma Sample Mode 2
3	Frame according to Figure E-1 Chroma Sample Mode 3

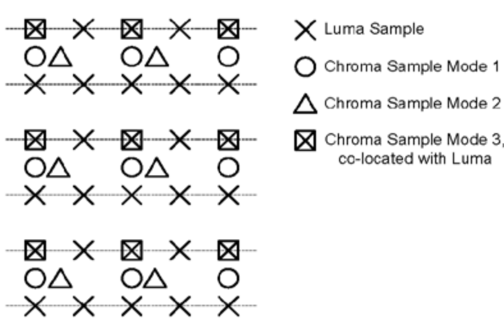

Figure E-1 – Luma and chroma sample types

chroma_location_field specifies the 4:2:0 sampling structure according to table E-7 and Figure E-2.

Table E-7 – Chroma Sampling Structure Frame

Value	Sampling Structure
0	undefined
1	Frame according to Figure E-2 Chroma Sample Mode 1
2	Frame according to Figure E-2 Chroma Sample Mode 2
3	Frame according to Figure E-2 Chroma Sample Mode 3

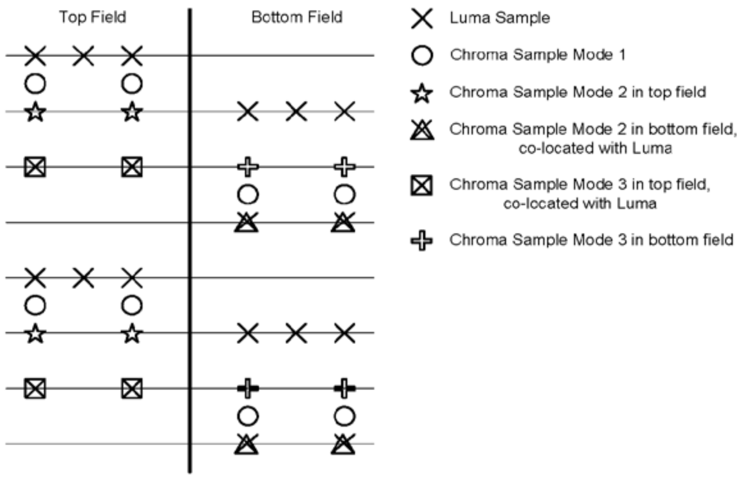

Figure E-2 – Luma and chroma association

timing_information_flag: A flag that, when 1, signals the presence of time unit information. If timing_information_flag is set to 0, then the following default values shall apply: num_units_in_tick = 0, time_scale = 0, fixed_frame_rate = 0.

num_units_in_tick is the number of time units of a clock operating at the frequency time_scale Hz that corresponds to one increment of a clock tick counter. A clock tick is the minimum interval of time that can be represented in the coded data. For example, if the clock frequency of a video signal is (30 000) ÷ 1001 Hz, time_scale may be 30 000 and num_units_in_tick may be 1001. If num_units_in_tick is 0, the duration of the clock tick is unspecified.

time_scale is the number of time units which pass in one second. For example, a time coordinate system that measures time using a 27 MHz clock has a time_scale of 27 000 000. If time_scale is 0, the duration of the clock tick specified above is unspecified.

fixed_frame_rate_flag: A bit that, if equal to 1, indicates that the temporal distance between the HRD output times of any two consecutive frames or fields in output order as defined in Annex C is a constant. If equal to 0, the temporal distances between HRD output times of consecutive frames or fields in output order as defined in Annex C may not be constant.

nal_hrd_flag: If nal hrd flag == '1', the multiplexed NAL and VCL stream complies with a hypothetical reference decoder (HRD) as specified in Annex C. In this case, the HRD parameters follow the nal_hrd_flag in the sequence parameter set syntax. If nal_hrd_flag == '0', the multiplexed NAL and VCL stream is not guaranteed to comply with an HRD. No default values are specified.

NOTE - If nal_hrd_flag == 0 the maximum buffer sizes and bit rates specified in Annex A apply.

vcl_hrd_flag: If vcl_hrd_flag == '1', the VCL bitstream complies with a hypothetical reference decoder (HRD) as specified in Annex C. In this case, the HRD parameters follow the vcl_hrd_flag in the sequence parameter set syntax. If vcl_hrd_flag == '0', the VCL bitstream is not guaranteed to comply with an HRD.

NOTE - If vcl_hrd_flag == 0 the maximum buffer sizes and bit rates specified in Annex A apply.

removal_time_tolerance: This syntax element indicates the number of clock ticks (see Annex C) of deviation allowed between the pre-decoder buffer removal times (see subclauses C.2.5 and C.3.5) and the accumulated Buffering Period capture time (see subclauses C.2.4 and C.3.4). It is encoded as a universal VLC, with all values allowed. A value of '0' implies that, at each measurement point (*i.e.* each picture preceding a Buffering Period SEI message), the removal time shall exactly match the capture time.

bitstream_restriction_flag: A flag that, when 1, signals the presence of bitstream restriction information. If bitstream_restriction_flag is set to 0, then the following default values shall apply: motion_vectors_over_pic_boundaries_flag = 1, minimum_compression_per_pic_reversed = 4, minimum_compression_per_macroblock_reversed = 1, log2_maximum_mv_length_vertical = 16, log2_maximum_mv_length_horizontal = 16.

motion_vectors_over_pic_boundaries_flag equal to 0 indicates that no motion vector refers to samples outside the picture boundaries. motion_vectors_over_pic_boundaries_flag equal to 1 indicates that motion vectors may refer to samples outside the picture boundaries.

minimum_compression_per_pic_reversed and **minimum_compression_per_macroblock_reversed** advise the decoder about the minimum compression ratio (corresponding to a maximum coded picture or macroblock size respectively). A value of n for either of the two indicates a minimum compression ratio of 1:n. Annex A defines the numbering range for both values.

log2_maximum_mv_length_vertical and **log2_maximum_mv_length_horizontal** indicate the maximum value of the absolute of a non-predicted vertical or horizontal motion vector component, in units of either ¼ or 1/8 sample, depending on the value of motion_vector_resolution. A value of n asserts that no absolute value of a motion vector component is bigger than 2**n ¼ pel or 1/8th pel units. Note: the high default value is restricted in Annex A for some profile/level combinations. Furthermore, the maximum vector length is restricted by the picture size.

E.3.2 HRD parameters semantics

pdb_cnt: This syntax element indicates the number of pre-decoder buffers (PBDs) in the HRD. A value of pdb_cnt equal to '0' is not allowed.

bit_rate_scale: Together with bit_rate_value[k], this syntax element defines the maximum input bit rate of the k-th PDB in an HRD.

bit_rate_value[k]: Together with bit_rate_scale, this syntax element defines the maximum input bit rate of the k-th PDB in an HRD. The actual bit rate in bits per second is given by:

$$\text{bit\_rate}[\,k\,] = \text{bit\_rate\_value}[k] * 2^{(6\,+\,\text{bit\_rate\_scale})}. \quad \text{(E-1)}$$

coded_pic_buffer_size_value is used together with coded_pic_buffer_size_scale[k] to define the maximum input bit rate of the k-th PDB in an HRD.

coded_pic_buffer_size_scale[k] is used together with coded_pic_buffer_size_value to define the pre-decoder buffer size of the k-th PDB in an HRD. The actual buffer size in bits is given by

$$\text{coded\_pic\_buffer\_size}[k] = \text{coded\_pic\_buffer\_size\_value}[k] * 2^{(4\,+\,\text{coded\_pic\_buffer\_size\_scale})}. \quad \text{(E-2)}$$

vbr_cbr_flag: If equal to '0', this syntax element indicates that the pre-decoder buffer operates in variable bit rate (VBR) mode. If equal to '1', it indicates constant bit rate (CBR) operation.

low_delay_hrd: If low_delay_hrd is equal to '0', the HRD operates in delay-tolerant mode. If low_delay_hrd is equal to '1', the HRD operates in low-delay mode. In low-delay mode, only one HRD buffer may be selected and big pictures which violate the HRD removal time rules at the pre-decoder buffer are permitted. It is expected that such big pictures occur only occasionally, but not mandatory.

E.3.3 VUI picture parameters semantics

frame_cropping_flag when 1, signals the presence of bitstream restriction information. If frame_cropping_flag is set to 0, then the following default values shall apply frame_cropping_rect_left = 0, frame_cropping_rect_right = 0, frame_cropping_rect_top = 0, frame_cropping_rect_bottom = 0.

frame_cropping_rect_left, frame_cropping_rect_right, frame_cropping_rect_top, frame_cropping_rect_bottom define the area of the luma picture internal array which shall be the output of the decoding process. The decoded values of these offsets consist of non-negative integer values, and the output of the decoding process is defined as the area within the rectangle containing luma samples with horizontal coordinates from cropping_rect_left to 16*(pic_width_in_mbs_minus1 + 1)-(cropping_rect_right + 1) and with vertical coordinates from cropping_rect_top to 16*(pic_height_in_mbs_minus1 + 1)-(cropping_rect_bottom + 1), inclusive.